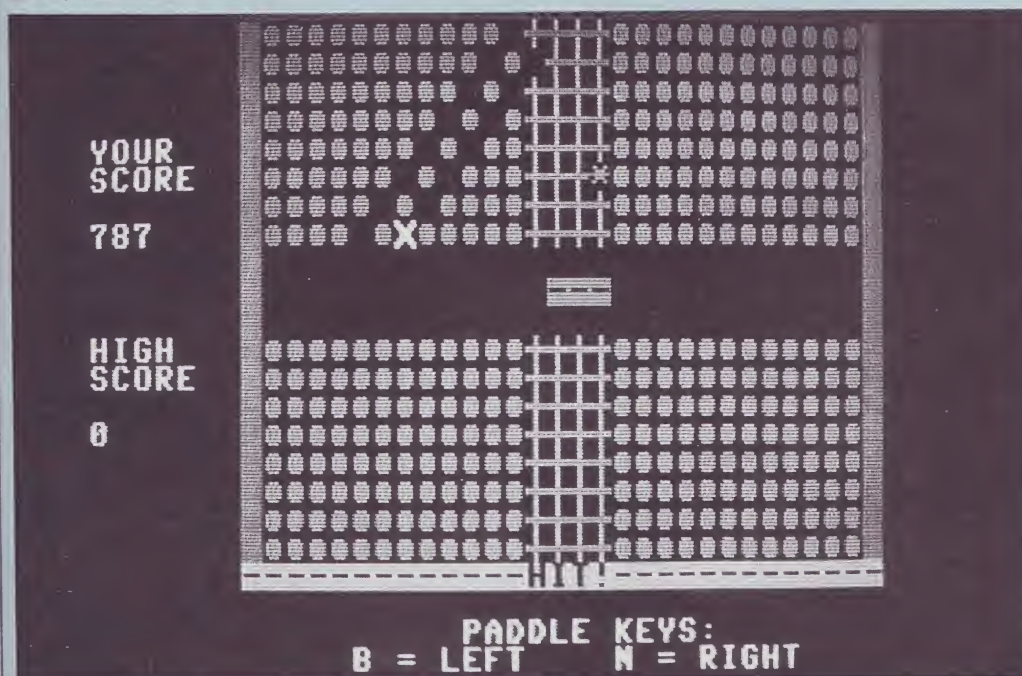


# HIT OR MISS

BY JOEY LATIMER



Commodore 64 version of *Hit or Miss*

*Hit or Miss* is an arcade-style game that's kind of a mixture of Ping-Pong, pin-ball, and bowling. When you first type RUN, you're presented with a square grid, surrounded by four rails. The upper and lower parts of the grid are filled with little circles (point markers) and plus signs (bumpers). The top and bottom rails are marked HIT and MISS, respectively.

Press any key to start the game. A projectile character will be sent into play, heading upwards at an angle toward the HIT rail. Your paddle will appear at the center line of the grid—move it left with the "B" key, right with the "N" key.

As the projectile travels upward, it encounters some of the circles or plus signs. Circles are worth 20 points each—they disappear when you hit them and do not change your projectile's direction. Plus signs also are worth 20 points, but these will send your projectile in a new direction before evaporating.

When the projectile encounters the HIT rail, it bounces off, and starts heading back. Every time

the projectile strikes the HIT rail, you receive a bonus equal to the number of pluses and circles you've already hit. The object of the game is to keep the projectile in play, and win points by preventing it from getting past your paddle and hitting the MISS rail. If it does, the game is over. Simple.

But wait! The HIT and MISS rails frequently swap position. You'll have to be ready to change your strategy in a split-second to keep the projectile away from the deadly trap.

And that's not all. Bonus markers (asterisks) appear randomly in the grid throughout the game. These are worth 500 points each. To further stir things up, more plus-sign bumpers will start appearing before you've played too long. A nest of plus signs can really bounce you around!

## PROGRAM NOTES

When you sit down to write an arcade-style game in BASIC, you have to make some hard decisions. BASIC is great for most general-purpose computing

projects, but it's slow, and this works against you when you want to write an action-oriented game.

Machine language, in the hands of a good programmer/designer, is fast enough to make almost any game scenario into a complex, delicately timed, and briskly responsive program. BASIC isn't—so the BASIC programmer has to plan an action-oriented game carefully to exploit BASIC's advantages and ensure that the game doesn't rely purely on speed for its appeal.

In designing *Hit or Miss* to run on six different computers, I took a careful look at BASIC, and reached the following conclusions. Since drawing complex graphics on the screen is not BASIC's strong suit, I knew I couldn't design an entertaining game that depended heavily on "realistic" images—spaceships, airplanes, and so on. On the other hand, one of the things BASIC does best and fastest is handle text; and letters, numbers, and graphic characters can combine to form interesting images. So I decided my game would be pro-

grammed in text mode, and that it would be based on simple, abstract shapes.

Next, I realized that since text mode provides only a very coarse array of screen positions, objects in my game would have to move in very simple ways—up, down, sideways, or diagonally. I decided on a bounce-and-ricochet theme, first because I could simulate these motions with simple movements, and second because the bouncing projectile could be formed from a single character. A paddle is a natural tool for bouncing things, and I had block characters to make it with—perfect! My theme would be single-paddle bounce-and-ricochet.

But what fun is bouncing, anyway? To provide a challenge, I had to find a reason for bouncing—some goal to be accomplished, or peril to be avoided. Eventually, I found both—but by a very indirect route. Hoping to provide a goal consistent with the scenario, I started by adding a group of single-character "markers" to the paddle and projectile display. When the projectile passed over these objects, the player would win points. Next, I hit on the idea that some of these markers also could act as "bumpers," changing the projectile's direction. The pattern of markers and bumpers added some interest to the display, as well. The variety of the whole scenario could be increased by having some bumpers and markers appear randomly as the game went on.

But then a couple of problems cropped up. At this point, the game was played across the entire screen, which was a long way to move a paddle in BASIC. The bumpers introduced so much randomness into the projectile's direction that I knew players would have problems anticipating what it would







```

340 IF PE=42 THEN SC=SC+500:C=15:GOSUB 3000
350 GOSUB 2000:IF PE=19 THEN DC=-DC:GOTO 460
360 IF CX>8 AND CX<37 THEN 390
370 DC=-DC:C=100:GOSUB 3000:CX=CX+DC
380 LOCATE CX,RX,PE:POSITION CX,RX:PUT #6,PE
390 IF RX>1 AND RX<21 AND PE<>146 THEN 460
400 DR=-DR:C=200:GOSUB 3000
410 IF RX<2 THEN RX=3:IF NOT FL THEN 570
420 IF RX>20 THEN RX=19:IF FL THEN 570
430 IF PE=146 THEN RX=RX+2*DR:GOTO 460
440 SC=SC+M:GOSUB 2000
450 IF RND(0)>0.7 THEN GOSUB 1000
460 POSITION HC,HR:PRINT S$;
470 POSITION CX,RX:PRINT Q$;
480 HC=CX:HR=RX:IF RX=11 THEN 270
490 POKE 555,1:POKE 753,0:K=PEEK(764):POKE 764,255:SOUND
0,0,0
500 NP=NP+2*((K=35)-(K=21))
510 IF HP=NP THEN 270
520 IF NP<9 THEN NP=9
530 IF NP>34 THEN NP=34
540 POSITION HP,11:PRINT N$;
550 POSITION NP,11:PRINT P$;
560 HP=NP:GOTO 270
570 FOR DE=1 TO 25:SOUND 0,INT(RND(0)*255),10,8
580 POSITION HC,HR:PRINT CHR$(INT(RND(0)*4)+42):NEXT DE:
SOUND 0,0,0
590 FOR DE=1 TO 30:PRINT :NEXT DE
600 PRINT "SORRY, YOU MISSED."
610 PRINT :PRINT "YOUR SCORE WAS ";SC;" POINTS."
620 IF SC>HS THEN HS=SC:PRINT :PRINT "A NEW RECORD!"
630 PRINT :PRINT "THE HIGH SCORE IS ";HS;" POINTS."
640 PRINT :PRINT :PRINT " PLEASE SELECT:":PRINT
650 PRINT " <R>EPLAY"
660 PRINT " <Q>UIT"
670 PRINT :PRINT :PRINT :PRINT
680 GET #1,K:IF K=81 THEN POKE 752,0:POKE 82,2:END
690 IF K<>82 THEN 680
700 GOTO 130
1000 SOUND 0,100,10,10:FL= NOT FL:IF FL THEN 1030
1010 POSITION 8,1:PRINT M$
1020 POSITION 8,21:PRINT H$:RETURN
1030 POSITION 8,1:PRINT HS
1040 POSITION 8,21:PRINT M$:RETURN
2000 POSITION 1,9:PRINT SC::RETURN
3000 SOUND 0,C,10,10:FOR S=1 TO 10:NEXT S:RETURN

```

### Commodore 64/Hit or Miss

```

10 SN=54272:SB=1024:CB=55296:HS=0:SD=0:R=214
20 POKE 650,128:POKE 53281,0:POKE 53280,0
30 FOR X=SN TO SN+24:POKE X,0:NEXT X
40 POKE SN+5,128:POKE SN+6,128:POKE SN+24,15
50 S$=CHR$(32):L$=S$:FOR X=1 TO 22:L$=L$+S$:NEXT X
60 B$=CHR$(113):FOR X=2 TO 28
70 B$=B$+CHR$(113-10*(X>12 AND X<17)):NEXT X
80 M$=CHR$(18)+"//////////MISS//////////"
90 H$=CHR$(18)+"=====HIT!=====
100 Q$=CHR$(5)+CHR$(118):E$=CHR$(144)+S$
110 P$=CHR$(18)+CHR$(156)+"---":N$=CHR$(144)+S$+S$+S$
120 CX=INT(RND(1)*25)+9:IF CX>20 AND CX<25 THEN 120
130 RX=11:FL=-1:DR=-1:DC=-1:SC=0
140 HC=CX:HR=RX:NP=22:LP=19:M=0
150 PRINT CHR$(147):POKE SN,198:POKE SN+1,45
160 POKE R,5:PRINT:PRINT CHR$(5);" YOUR":PRINT " SCORE:"
:PRINT:PRINT SC
170 POKE R,12:PRINT:PRINT " HIGH":PRINT " SCORE:"
180 PRINT:PRINT CHR$(5);HS
190 POKE R,10:PRINT:PRINT TAB(12);"PRESS ANY KEY TO BEGI
N.":CHR$(19)
200 K1=INT(RND(1)*3)+152:K2=INT(RND(1)*7)+149:K0=K1
210 PRINT:FOR X=1 TO 19:PRINT TAB(8);CHR$(18);CHR$(156);
S$;
220 IF X<9 OR X>11 THEN PRINT CHR$(146);CHR$(K0);B$;
230 PRINT TAB(37);CHR$(18);CHR$(156);S$
240 IF X=11 THEN K0=K2
250 NEXT X

```

```

260 POKE R,22:PRINT:PRINT TAB(18);"PADDLE KEYS:"
270 PRINT TAB(13);"B = LEFT      N = RIGHT";CHR$(19)
280 GOSUB 1030
290 GET K$:IF K$="" THEN 290
300 POKE R,10:PRINT:PRINT TAB(12);L$
310 IF RND(0)>0.05 THEN 360
320 CH=42:IF M>300 AND RND(1)>0.5 THEN CH=123
330 XR=INT(RND(1)*16)+3
340 IF XR>9 AND XR<13 THEN 330
350 POKE R,XR-1:PRINT:PRINT TAB(INT(RND(1)*21)+10);CHR$(
5);CHR$(CH)
360 CX=CX+DC:RX=RX+DR
370 PE=PEEK(SB+CX+RX*40)
380 IF PE=81 OR PE=91 THEN SC=SC+20:M=M+1
390 IF PE=42 THEN SC=SC+500:SD=33:POKE SN+4,SD
400 GOSUB 2000:IF PE=91 THEN DC=-DC:GOTO 520
410 IF CX>8 AND CX<37 THEN 440
420 DC=-DC:SD=129:POKE SN+4,SD:CX=CX+DC
430 PE=PEEK(SB+CX+RX*40)
440 IF RX>1 AND RX<21 AND PE<>173 THEN 520
450 DR=-DR:SD=129:POKE SN+4,SD
460 IF RX<2 THEN RX=3:IF NOT FL THEN 620
470 IF RX>20 THEN RX=19:IF FL THEN 620
480 IF PE=173 THEN RX=RX+2*DR:GOTO 520
490 IF RX=11 THEN PRINT PE:STOP
500 SC=SC+M:GOSUB 2000
510 IF RND(1)>0.7 THEN GOSUB 1000
520 POKE R,HR-1:PRINT:PRINT TAB(HC);S$
530 POKE R,RX-1:PRINT:PRINT TAB(CX);Q$
540 HC=CX:HR=RX:IF RX=11 THEN 310
550 GET K$:NP=NP+2*((K$="B")-(K$="N")):IF SD THEN POKE S
N+4,SD-1:SD=0
560 IF LP=NP THEN 310
570 IF NP<9 THEN NP=9
580 IF NP>34 THEN NP=34
590 POKE R,10:PRINT:PRINT TAB(LP);N$
600 POKE R,10:PRINT:PRINT TAB(NP);P$
610 LP=NP:GOTO 310
620 POKE SN+4,33:FOR DE=1 TO 25
630 POKE R,HR-1:PRINT:PRINT TAB(HC);CHR$(INT(RND(1)*2)+1
18)
640 POKE SN,38:POKE SN+1,INT(RND(1)*69)+1:NEXT DE
650 POKE SN+4,0
660 POKE R,23:PRINT
670 FOR DE=1 TO 10:PRINT CHR$(13):NEXT DE
680 PRINT CHR$(158);"SORRY, YOU MISSED."
690 PRINT:PRINT "YOUR SCORE WAS";SC;"POINTS."
700 IF SC>HS THEN HS=SC:PRINT:PRINT CHR$(159);"A NEW REC
ORD!":CHR$(158)
710 PRINT:PRINT "THE HIGH SCORE IS";HS;"POINTS."
720 PRINT:PRINT:PRINT:PRINT CHR$(150);" PLEASE SELECT:":
PRINT
730 PRINT " <R>EPLAY"
740 PRINT " <Q>UIT"
750 PRINT:PRINT:PRINT:PRINT
760 GET K$:IF K$="Q" THEN END
770 IF K$<>"R" THEN 760
780 GOTO 120
1000 POKE SN+4,17:FL=NOT FL:IF FL THEN 1030
1010 POKE R,1:PRINT:PRINT CHR$(19);CHR$(28);TAB(48);M$
1020 POKE R,20:PRINT:PRINT CHR$(158);TAB(8);H$:RETURN
1030 POKE R,1:PRINT:PRINT CHR$(19);CHR$(158);TAB(48);H$
1040 POKE R,20:PRINT:PRINT CHR$(28);TAB(8);M$:RETURN
2000 POKE R,8:PRINT:PRINT CHR$(5);SC:RETURN

```

### IBM PCs/Hit or Miss

```

10 CLS:KEY OFF:SCREEN 0,1:WIDTH 40:LOCATE ,,0
20 HS=0:V$=CHR$(176):S$=CHR$(32)
30 L$=S$:FOR X=1 TO 22:L$=L$+S$:NEXT X
40 FOR X=1 TO 28
50 B$=B$+CHR$(43-((X<13)+(X>16))*36):NEXT X
60 M$="//////////MISS//////////"
70 H$="=====HIT!=====
80 Q$=CHR$(127):P$=STRING$(4,219):N$=STRING$(4,32)
90 CX=INT(RND*25)+9:IF CX>20 AND CX<25 THEN 90
100 FL=-1:DR=-1:DC=-1:SC=0:RX=11

```



# ARCADE GAME

```

110 HC=CX:HR=RX:NP=21:LP=19:M=0
120 CLS
130 COLOR 3:LOCATE 6,1:PRINT "YOUR":PRINT "SCORE":PRINT SC
140 COLOR 10:LOCATE 13,1:PRINT "HIGH":PRINT "SCORE":PRINT HS
150 KO=3:FOR X=2 TO 20:LOCATE X,8:COLOR 5:PRINT V$;
160 IF X>9 AND X<13 THEN PRINT SPC(28);V$:GOTO 180
170 COLOR KO:PRINT B$;:COLOR 5:PRINT V$
180 IF X=12 THEN KO=2
190 NEXT X
200 COLOR 15:LOCATE 11,12:PRINT "PRESS ANY KEY TO BEGIN."
210 COLOR 7:LOCATE 23,17:PRINT "PADDLE KEYS:"
220 PRINT TAB(12);"B = LEFT    N = RIGHT";
230 GOSUB 1030
240 K$=INKEY$:IF K$="" THEN 240
250 LOCATE 11,12:PRINT L$
260 IF RND>.05 THEN 310
270 CH=42:IF M>300 AND RND>.5 THEN CH=43
280 XR=INT(RND*16)+3
290 IF XR>9 AND XR<13 THEN 280
300 LOCATE XR,INT(RND*21)+10:COLOR 7:PRINT CHR$(CH);
310 CX=CX+DC:RX=RX+DR
320 PE=SCREEN(RX,CX,0)
330 IF PE=43 OR PE=79 THEN SC=SC+20:M=M+1:GOSUB 2000
340 IF PE=42 THEN SOUND 550,1:SC=SC+500:GOSUB 2000
350 IF PE=43 THEN DC=-DC:GOTO 460
360 IF CX>8 AND CX<37 THEN 390
370 DC=-DC:SOUND 100,5
380 CX=CX+DC:PE=SCREEN(RX,CX,0)
390 IF RX>1 AND RX<21 AND PE<219 THEN 460
400 DR=-DR:SOUND 90,5
410 IF RX<2 THEN RX=2:IF NOT FL THEN 570
420 IF RX>20 THEN RX=20:IF FL THEN 570
430 IF PE=219 THEN RX=RX+2*DR:GOTO 460
440 SC=SC+M:GOSUB 2000
450 IF RND>.7 THEN GOSUB 1000
460 LOCATE HR,HC:PRINT S$;
470 LOCATE RX,CX:COLOR 15:PRINT Q$:COLOR 7
480 HC=CX:HR=RX:IF RX=11 THEN 260
490 K$=INKEY$
500 NP=NP+((K$="B")-(K$="N"))*2
510 IF LP=NP THEN 260
520 IF NP<9 THEN NP=9
530 IF NP>33 THEN NP=33
540 LOCATE 11,LP:PRINT N$:LP=NP
550 LOCATE 11,NP:PRINT P$
560 GOTO 260
570 FOR DE=1 TO 45:SOUND 300,.2:SOUND 800,.4
580 LOCATE HR,HC:PRINT CHR$(INT(RND*2)+42)
590 NEXT DE:LOCATE 23,1
600 FOR DE=1 TO 10:FOR DL=1 TO 20:NEXT DL
610 PRINT:NEXT DE:PRINT "SORRY, YOU MISSED."
620 PRINT:PRINT "YOUR SCORE WAS";SC;"POINTS."
630 IF SC>HS THEN HS=SC:PRINT:PRINT "A NEW RECORD!"
640 PRINT:PRINT "THE HIGH SCORE IS";HS;"POINTS."
650 PRINT:PRINT:PRINT:PRINT " PLEASE SELECT:":PRINT
660 PRINT " <R>EPLAY":PRINT " <Q>UIT"
670 PRINT:PRINT:PRINT
680 K$=INKEY$:IF K$="Q" THEN END
690 IF K$="R" THEN 90 ELSE 680
1000 SOUND 400,2:SOUND 800,3:FL=NOT FL:IF FL THEN 1030
1010 LOCATE 1,8:COLOR 5:PRINT V$;:COLOR 4:PRINT M$;:COLOR 5:PRINT V$
1020 LOCATE 21,8:PRINT V$;:COLOR 14:PRINT H$;:COLOR 5:PRINT V$:RETURN
1030 LOCATE 1,8:COLOR 5:PRINT V$;:COLOR 14:PRINT H$;:COLOR 5:PRINT V$
1040 LOCATE 21,8:PRINT V$;:COLOR 4:PRINT M$;:COLOR 5:PRINT V$:RETURN
2000 COLOR 3:LOCATE 9,1:PRINT SC;:RETURN

```

## TRS-80 Color Computer/Hit or Miss

```

10 CLEAR 1000:HS=0:S$=CHR$(32)
20 L$=S$:FOR X=1 TO 21:L$=L$+S$:NEXT X

```

```

30 B$="0":FOR X=2 TO 22
40 B$=B$+CHR$(43-((X<10)+(X>13))*36):NEXT X
50 A$=CHR$(159):FOR X=1 TO 8:A$=A$+CHR$(159):NEXT X
60 H$=CHR$(255)+A$+CHR$(104)+CHR$(105)+CHR$(116)+CHR$(127)+A$+CHR$(255)
70 A$=CHR$(191):FOR X=1 TO 8:A$=A$+CHR$(191):NEXT X
80 M$=CHR$(255)+A$+CHR$(109)+CHR$(105)+CHR$(115)+CHR$(115)+A$+CHR$(255)
90 Q$="X"
100 P$=CHR$(175)+CHR$(175)+CHR$(175)+CHR$(175):N$=S$+S$+S$+S$
110 CX=RND(20)+8:IF CX>16 AND CX<20 THEN 110
120 RX=8:FL=-1:DR=-1:DC=-1:SC=0
130 HC=CX:HR=RX:NP=17:HP=16:M=0
140 CLS
150 PRINT@64,"YOUR":PRINT "SCORE":PRINT:PRINT SC
160 PRINT@288,"HIGH":PRINT "SCORE":PRINT:PRINT HS
170 PRINT@232,"PRESS ANY KEY TO BEGIN";
180 FOR X=1 TO 13
190 PRINT@7+X*32,CHR$(255);
200 IF X>5 AND X<9 THEN 220
210 PRINT B$;
220 PRINT@30+X*32,CHR$(255)
230 NEXT X
240 PRINT@490,"B=LEFT    N=RIGHT";
250 GOSUB 1030
260 K$=INKEY$:IF K$="" THEN 260
270 PRINT@232,L$;
280 IF RND(10)>.5 THEN 320
290 CH=42:IF M>150 AND RND(10)>.5 THEN CH=43
300 XR=RND(13):IF XR>5 AND XR<9 THEN 300
310 XC=RND(22)+7:PRINT@XC+XR*32,CHR$(CH);
320 CX=CX+DC:RX=RX+DR:PE=PEEK(1024+CX+RX*32)
330 IF PE=107 OR PE=79 THEN SC=SC+20:M=M+1:PRINT@160,S C;
340 IF PE=106 THEN SC=SC+500:PRINT@160,SC;:SOUND 50,1
350 IF PE=107 THEN DC=-DC:GOTO 460
360 IF CX>8 AND CX<30 THEN 390
370 DC=-DC:CX=CX+DC:SOUND 200,1
380 PE=PEEK(1024+XC+XR*32)
390 IF RX>0 AND RX<14 AND PE<175 THEN 460
400 DR=-DR:SOUND 244,1
410 IF RX<1 THEN RX=2:IF NOT FL THEN 590
420 IF RX>13 THEN RX=12:IF FL THEN 590
430 IF PE=175 THEN RX=RX+2*DR:GOTO 460
440 SC=SC+M:PRINT@160,SC;
450 IF RND(10)>.7 THEN GOSUB 1000
460 PRINT@HC+HR*32,S$;
470 PRINT@CX+RX*32,Q$;
480 HC=CX:HR=RX:IF RX=7 THEN 280
489 REM --TO USE JOYSTICK, REMOVE REM FROM LINE 490--
490 REM JO=JOYSTK(0)
499 REM --TO USE JOYSTICK, REMOVE REM FROM LINE 500--
500 REM NP=NP-3*((JO>20)-(JO<43))
510 K$=INKEY$
520 NP=NP+3*((K$="B")-(K$="N"))
530 IF NP=HP THEN 280
540 IF NP<8 THEN NP=8
550 IF NP>26 THEN NP=26
560 PRINT@HP+7*32,N$;
570 PRINT@NP+7*32,P$;:HP=NP
580 GOTO 280
590 FOR DE=1 TO 25:SOUND RND(244),1:PRINT@HC+HR*32,CHR$(RND(110)+128);:NEXT DE
600 PRINT@478," ":FOR L=1 TO 16:PRINT:FOR DE=1 TO 35:NEXT DE:NEXT L
610 PRINT "SORRY, YOU MISSED."
620 PRINT:PRINT "YOUR SCORE WAS";SC;"POINTS."
630 IF SC>HS THEN HS=SC:PRINT:PRINT "A NEW RECORD!"
640 PRINT:PRINT "THE HIGH SCORE IS";HS;"POINTS."
650 PRINT:PRINT:PRINT:PRINT " PLEASE SELECT:":PRINT
660 PRINT " <R>EPLAY"
670 PRINT " <Q>UIT"
680 PRINT
690 K$=INKEY$:IF K$="Q" THEN END
700 IF K$="R" THEN 110 ELSE 690
1000 SOUND 2,2:FL=NOT FL:IF FL THEN 1030

```



```

1010 PRINT@7,M$;
1020 PRINT@455,H$;:RETURN
1030 PRINT@7,H$;
1040 PRINT@455,M$;:RETURN

```

### TRS-80 Model III/Hit or Miss

```

10 CLEAR 1000:CLS:HS=0:SS=CHR$(32):V$=CHR$(191)
20 L$=SS:FOR X=1 TO 22:L$=L$+SS:NEXT X
30 B$=V$:FOR X=1 TO 28
40 B$=B$+CHR$(43-((X<13)+(X>16))*68):NEXT X
50 M$=V$+//////////MISS//////////+V$
60 H$=V$+=====HIT!=====+V$
70 B$=B$+V$:Q$="X":P$=STRING$(4,179):N$=STRING$(4,32)
80 CX=RND(25)+17:IF CX>20 AND CX<25 THEN 80
90 RX=8:FL=-1:DR=-1:DC=-1:SC=0
100 HC=CX:HR=RX:NP=28:LP=20:M=0
110 CLS
120 PRINT@128,"YOUR":PRINT "SCORE":PRINT:PRINT SC
130 PRINT@576,"HIGH":PRINT "SCORE":PRINT:PRINT HS
140 FOR X=1 TO 13
150 IF X>5 AND X<9 THEN PRINT@16+X*64,V$;TAB(45);V$;:G
OTO 170
160 PRINT@16+64*X,B$;
170 NEXT X:PRINT@468,"PRESS ANY KEY TO BEGIN.";
180 PRINT@435,"PADDLE KEYS:";
190 PRINT@561,"B=LEFT N=RIGHT";
200 GOSUB 1020
210 K$=INKEY$:IF K$="" THEN 210
220 PRINT @468,L$;
230 IF RND(20)>1 THEN 280
240 CH=42:IF M>200 AND RND(10)>5 THEN CH=19
250 XR=RND(12)+1
260 IF XR>5 AND XR<9 THEN 250
270 PRINT@RND(27)+17+XR*64,CHR$(CH);
280 CX=CX+DC:RX=RX+DR
290 PE=PEEK(15360+CX+RX*64)
300 IF PE=43 OR PE=111 THEN SC=SC+20:M=M+1:GOSUB 2000
310 IF PE=42 THEN SC=SC+500:GOSUB 2000
320 IF PE=43 THEN DC=-DC:GOTO 430
330 IF CX>16 AND CX<44 THEN 360
340 DC=-DC:CX=CX+DC
350 PE=PEEK(15360+CX+RX*64)
360 IF RX>0 AND RX<14 AND PE<>179 THEN 430
370 DR=-DR
380 IF RX<1 THEN RX=1:IF NOT FL THEN 540
390 IF RX>13 THEN RX=13:IF FL THEN 540
400 IF PE=179 THEN RX=RX+2*DR:GOTO 430
410 SC=SC+M:GOSUB 2000
420 IF RND(10)>7 THEN GOSUB 1000
430 PRINT@HC+HR*64,S$;
440 PRINT@CX+RX*64,Q$;
450 HC=CX:HR=RX:IF RX=7 THEN 230
460 K$=INKEY$
470 NP=NP+((K$="B")-(K$="N"))*2
480 IF LP=NP THEN 230
490 IF NP<17 THEN NP=17
500 IF NP>41 THEN NP=41
510 PRINT@LP+448,N$;
520 PRINT@NP+448,P$;:LP=NP
530 GOTO 230
540 FOR DE=1 TO 50:PRINT@HC+HR*64,CHR$(RND(3)+32);
550 NEXT DE:PRINT@960,""
560 FOR L=1 TO 16:PRINT:FOR DE=1 TO 5:NEXT DE:NEXT L
570 PRINT "SORRY, YOU MISSED."
580 PRINT:PRINT "YOUR SCORE WAS";SC;"POINTS."
590 IF SC>HS THEN HS=SC:PRINT "A NEW RECORD!"
600 PRINT "THE HIGH SCORE IS";HS;"POINTS."
610 PRINT:PRINT:PRINT:PRINT "PLEASE SELECT:":PRINT
620 PRINT "<R>EPLAY":PRINT "<Q>UIT":PRINT
630 K$=INKEY$:IF K$="Q" THEN END
640 IF K$="R" THEN 80 ELSE 630
1000 FL=NOT FL:IF FL THEN 1020
1010 PRINT@16,M$;:PRINT@912,H$;:RETURN
1020 PRINT@16,H$;:PRINT@912,M$;:RETURN
2000 PRINT@320,SC;:RETURN

```

## FEATURE PROGRAM

# PGRAPH!

BY MARK GILMAN



ATARI  
PLAYER/  
MISSILE  
GRAPHICS  
MADE  
EASY

When the Statue of Liberty was a "young lady" in her 20s, she welcomed all four of my grandparents to America. She is 98 now—wind and time have exacted their toll. On July 4, 1984, her famous torch was lowered as part of a major overhaul. The general public will have to wait until July 4 of next year to see the torch raised aloft again. But *you* don't have to wait that long. With your joystick and our demonstration program, *Miss Liberty*, you can set the torch on high well ahead of schedule!

This snazzy special effect is produced with one of the Atari's "players" (referred to as "sprites" on other systems; see Dr. Kursor's Clinic, p. 59) and *PGraph!*, a utility package that makes Atari player/missile graphics easy to use. Player/missile (or "P/M") graphics may be one of the Atari's least understood, but most powerful, features.

Players are graphic objects that are completely independent of everything else on the screen, and "missiles" are narrow players, useful to represent things like bullets. Both can be superimposed on any screen and moved around rapidly without affecting what's displayed underneath. Special sensors detect when they collide with each other or with parts of the background. "Priorities" can be set so that players or missiles seem to move in front of, or behind, other things on the screen. For these reasons, P/M graphics are used heavily by Atari game, graphics, and animation programmers.

Unfortunately, Atari BASIC provides no direct way of using players or missiles. Let's take a closer look at how P/M graphics work.

### BIT-MAPPING

A simple way to think of a player is to visualize a grid, such as a narrow strip of graph paper, eight blocks wide by some number of blocks high. By coloring in some of the blocks on this grid and leaving others blank, you can make a picture: a spaceship, a happy face, etc.